AP CYBERSECURITY • EXERCISE SHEET

5.1 Application and Data Vulnerabilities and Attacks

Name: _____ Class: _____ Date: _____ **Total: 10 marks**

Objective

Build the skills to answer exam questions on application attacks such as **SQL injection** 数据库注入 and **directory traversal** 目录遍历.

You must be able to:

- identify an application attack from log evidence
- explain how unchecked user input enables injection attacks
- name **data validation** 数据验证 as the core defense

1 Worked examples

Study these first. Each one shows the method for a task used later.

Reading the evidence

SQL injection shows OR 1=1 and – in a query. XSS shows a <script> tag. Directory traversal shows ../ in a URL. A buffer overflow shows an unusually long input string.

The common cause

Every one of these attacks exploits unchecked user input. Data validation - checking input meets expected rules before processing - is the core defense.

2 Practice

2.1 A login field receives the input ' OR '1'='1. Name the attack. [1]

2.2 Explain how a **directory traversal** attack works. [2]

2.3 A web application does not check what users type into its input fields.

(a) State the general name for the resulting class of attack. [1]

(b) Name the defense that would prevent it. [1]

3 Exam-style questions

3.1 A URL request for ../../../etc/passwd is a [1]

- **A** SQL injection
- **B** XSS attack
- **C** directory traversal attack
- **D** smurf attack

3.2 Sending far more data than a memory buffer can hold is a [1]

- **A** buffer overflow
- **B** cross-site scripting
- **C** SQL injection
- **D** directory traversal

3.3 A bank's log shows an input string containing ' OR '1'='1' – in its loan form.

(a) Name the attack. [1]

(b) Explain what the attacker is trying to do. [2]

4 Go further

- work through the **5.1 Application and Data Vulnerabilities and Attacks** lesson on the **Learn** page;
- read the **Securing Applications and Data** section of the AP Cybersecurity handout on the **Know** page.

Solutions

2.1 SQL injection.

2.2 the attacker puts ../ sequences in a URL to climb out of the intended folder and reach sensitive files.

2.3 (a) injection attacks. (b) data validation / input sanitization.

3.1 C. ../ climbing folders is directory traversal.

3.2 A. overflowing a buffer into nearby memory is a buffer overflow.

3.3 (a) SQL injection. (b) inject control characters into the database query to make it return or change data it should not.